

PATENT

S/N unknown

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Akimitsu SHIMAMURA

Serial No.: unknown

Filed: concurrent herewith

Docket No.:

10873.785US01

Title: COMPUTER SYSTEM

CERTIFICATE UNDER 37 CFR 1.10

U.S. DEPARTMENT OF JUSTICE
FEDERAL BUREAU OF INVESTIGATION
WASHINGTON, D.C. 20535
TELEPHONE (202) 512-2400
FAX (202) 512-2400
E-MAIL: fbi@fbi.gov
WWW: www.fbi.gov
U.S. MAIL PERMIT NO. 1000 WASHINGTON, D.C. 20535
POSTAGE WILL BE PAID BY ADDRESSEE
FIRST CLASS
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES
E UNDER 37 CFR 1.10
'Express Mail' mailing label number: EL920771586US
JAN 26 2001

Date of Deposit: September 25, 2001

I hereby certify that this correspondence is being deposited with the United States Postal Service 'Express Mail Post Office To Addressee' service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

By:

Name: Omesh Singh

09/963894
09/25/01

SUBMISSION OF PRIORITY DOCUMENT(S)

Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

Applicants enclose herewith one certified copy of a Japanese application, Serial

No. 2000-292178, filed September 26, 2000, the right of priority of which is claimed under 35

U.S.C. § 119.

Respectfully submitted,

MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
(612) 332-5300

Dated: September 25, 2001

By

Douglas P. Mueller
Reg. No. 30,300

DPM/tvm/jlc

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年 9月26日

出 願 番 号

Application Number:

特願2000-292178

出 願 人

Applicant(s):

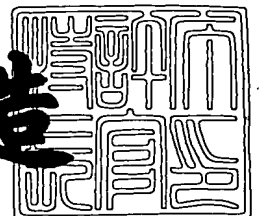
松下電器産業株式会社



2001年 5月31日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3049638

【書類名】 特許願

【整理番号】 R4457

【提出日】 平成12年 9月26日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/38

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内

【氏名】 島村 秋光

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100095555

【弁理士】

【氏名又は名称】 池内 寛幸

【電話番号】 06-6361-9334

【選任した代理人】

【識別番号】 100076576

【弁理士】

【氏名又は名称】 佐藤 公博

【選任した代理人】

【識別番号】 100107641

【弁理士】

【氏名又は名称】 鎌田 耕一

【選任した代理人】

【識別番号】 100110397

【弁理士】

【氏名又は名称】 席丘 圭司

【選任した代理人】

【識別番号】 100115255

【弁理士】

【氏名又は名称】 辻丸 光一郎

【選任した代理人】

【識別番号】 100115152

【弁理士】

【氏名又は名称】 黒田 茂

【手数料の表示】

【予納台帳番号】 012162

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0004605

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置

【特許請求の範囲】

【請求項 1】 パイプラインで制御される情報処理装置において、パイプラインが、論理演算器におけるクリティカルパスを正常に実行するためのクロック周波数より高速なクロック周波数で駆動され、

パイプラインの実行ステージにおいて、前記パイプラインのクロック周波数で駆動する高速駆動の論理演算器と、前記クリティカルパスを正常に実行するためのクロック周波数以下のクロック周波数で駆動する低速駆動の論理演算器を備え

前記高速駆動の論理演算器が正しく論理演算処理が実行できた場合は、当該論理演算結果をパイプライン実行ステージの出力とし、前記高速駆動の論理演算器が正しく論理演算処理が実行できなかった場合は、前記低速駆動の論理演算器の論理演算結果をパイプライン実行ステージの出力とすることを特徴とする情報処理装置。

【請求項 2】 前記低速駆動の論理演算器が、複数の低速駆動の論理演算器からなり、

実行ステージで用いる低速駆動の論理演算器を実行ステージごとに順番に切り替え、各低速駆動の論理演算器は、自分が担当した実行ステージの演算を前記クリティカルパスを正常に実行するためのクロック周波数以下のクロック周波数で処理する請求項 1 に記載の情報処理装置。

【請求項 3】 前記パイプラインのクロック周波数が、前記クリティカルパスを正常に実行するためのクロック周波数の n 倍である場合、前記複数の低速駆動の論理演算器を n 個の低速駆動の論理演算器とし、

パイプラインにおいて順に実行される n 個の実行ステージの演算を、 n 個の低速駆動の論理演算器が順に担当して行く請求項 2 に記載の情報処理装置。

【請求項 4】 同じ実行ステージの演算結果として得られた前記高速駆動の論理演算器の出力と前記低速駆動の論理演算器の出力とを入力として両者を比較する比較器を備え、

パイプライン実行ステージの出力処理にあたり、前記高速駆動の論理演算器の出力をパイプライン実行ステージの出力として仮定しておき、前記比較器における比較処理結果が一致を示す場合は、前記高速駆動の論理演算器の出力を実行ステージの出力として確定してパイプライン処理を継続し、前記比較器における比較処理結果が一致を示さない場合は、前記低速駆動の論理演算器の出力を実行ステージの出力として置き換える請求項 1 ～ 3 のいずれかに記載の情報処理装置。

【請求項 5】 前記比較器における比較処理結果が一致を示さない場合において、前記低速駆動の論理演算器の出力を実行ステージの出力として置き換える処理が完了するまで、パイプラインの全ステージの駆動を一時停止する請求項 4 に記載の情報処理装置。

【請求項 6】 一定期間の前記不一致検出信号の発生回数をカウントする回路と、前記カウント数によってパイプラインを駆動する周波数を変化させる回路を備えたことを特徴とする請求項 1 記載の情報処理装置。

【請求項 7】 前記パイプラインのクロック周波数を高めることによる前記高速駆動の論理演算器の処理量の増加と、前記パイプラインのクロック周波数を高めることにより前記高速駆動の論理演算器が正しく論理演算処理ができず、前記低速駆動の論理演算器の論理演算結果をパイプライン実行ステージの出力とする処理が増加することによる処理遅延量の増加とを比較し、前者が後者に対して所定割合より大きい場合にパイプラインのクロック周波数を上げる請求項 1 に記載の情報処理装置。

【請求項 8】 前記パイプラインのクロック周波数を下げることによる前記高速駆動の論理演算器の処理量の低下と、前記パイプラインのクロック周波数を下げることにより前記高速駆動の論理演算器が正しく論理演算処理ができ、前記低速駆動の論理演算器の論理演算結果をパイプライン実行ステージの出力とする処理が減少することによる処理遅延量の低下とを比較し、後者が前者に対して所定割合より大きい場合にパイプラインのクロック周波数を下げる請求項 1 に記載の情報処理装置。

【請求項 9】 複数の論理演算器と、クリティカルパスとなるテストデータを生成するデータ生成回路と、各論理演算器のクリティカルパス実行時間を測定す

る実行時間測定回路と、クリティカルパス実行時間が最短となる論理演算器を検出する検出回路を備え、

前記検出回路によって検出された論理演算器を前記高速駆動の論理演算器とし、他の一または複数の論理演算器を前記低速駆動の論理演算器として選択することを特徴とする請求項 1 記載の情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は情報処理装置におけるパイプライン制御に関するものである。

【0002】

【従来の技術】

図 10 は従来の情報処理装置のブロック図である。

【0003】

同図において従来の情報処理装置は、インストラクションフェッチブロック（以下 I F B と略記する。） 1 と、デコードブロック（以下 D E C B と略記する。） 4 と、実行ブロック（以下 E X B と略記する。） 7 とによって構成されている。各ブロックにはクロック信号 C L K が入力されている。I F B 1 内にはフェッチしたインストラクションを格納する命令レジスタ 2 があり、命令レジスタ 2 の出力 3 が D E C B 4 へ入力されている。D E C B 4 内には命令レジスタ 2 の出力 3 をデコードした信号を格納するデコード信号レジスタ 5 があり、デコード信号レジスタ 5 の出力 6 が E X B 7 へ入力されている。E X B 7 においては論理演算器（以下 A L U と略記する。） 20 へ入力データとして A L U A、A L U B が、制御信号として A L U C N T が接続され、A L U 20 の出力 23 は実行結果を格納する E X レジスタ 30 に入力されている。A L U A、A L U B、A L U C N T はデコード信号レジスタ 5 の出力 6 によって決定される。

【0004】

以上の構成を有する情報処理装置のパイプライン動作をタイミングチャートを用いて説明する。

【0005】

図 1 1 は従来の情報処理装置の動作を示すタイミングチャートである。

【0 0 0 6】

図中、a の期間では命令 1 のフェッチが I F B 1 で行われる。

【0 0 0 7】

次に、b の期間の開始により命令レジスタ 2 へ命令 1 の命令コードが格納される。また、b の期間において I F B 1 は命令 2 のフェッチを行い、D E C B 4 において命令レジスタ 2 の出力 3 をデコードした命令 1 のデコード信号が生成される。

【0 0 0 8】

次に、c の期間の開始で命令 1 のデコード信号がデコード信号レジスタ 5 へ格納される。また、c の期間において I F B 1 は命令 3 のフェッチを行い、D E C B 4 においては命令 2 のデコードを行い、E X B 7 においてデコード信号レジスタ 5 から出力される命令 1 のデコード信号によって、対象データ間の演算が行われる。

【0 0 0 9】

次に、d の区間の開始で E X レジスタ 3 0 へ命令 1 によって行われた演算結果が格納される。また、d の期間において I F B 1 は命令 4 のフェッチを行い、D E C B 4 において命令 3 のデコードを行い、E X B 7 においては命令 2 の演算が行われる。

【0 0 1 0】

以上のように、a の期間において命令 1 がメモリからフェッチされ、d の期間においてその演算結果が E X レジスタ 1 3 0 に格納される。この一連の処理をパイプラインとして次々に命令を実行して行く。

【0 0 1 1】

【発明が解決しようとする課題】

E X B 7 における演算実行時間は、演算されるデータの値に依存する。そのため、想定される処理のうちもっとも演算実行時間を要するケースが、処理高速化に対するボトルネックとなる。このもっとも演算実行時間を要するケースをクリティカルパスと呼ぶ。クリティカルパスは上記の通り、演算されるデータの値に

依存する。例えば、加減算時においては、演算時にキャリー伝播が最大になる場合がクリティカルパスとなる。

【0012】

従来の情報処理装置において、情報処理装置の正常動作を確保せしめるため、各ブロックへ供給されるクロックCLKの周波数はEXB7においてクリティカルパスが正常に実行できる周波数として設定せざるを得ない。つまり、クリティカルパスという最も厳しいケースを正常に処理できる程度にまでしかクロック周波数を高めることができなかった。

【0013】

クリティカルパスがどの程度の頻度で発生するかは、情報処理装置が実際に処理するデータにより決まるものであるが、クリティカルパスの発生頻度が低いことも十分に想定され、実際に処理する演算のほとんどがクロック周期よりも短い演算実行時間で終了してしまっているケースや、クリティカルパスが結果的に発生しないケースが多い。しかし、これらの場合でも、クリティカルパス発生の可能性を考慮するとクロック周期を短くすることが許されず、情報処理装置の処理性能向上の障害になっていた。

【0014】

本発明はかかる課題に鑑み、パイプラインをクリティカルパス実行可能クロックより短い周期のクロックで駆動し、かつ、クリティカルパス発生時の正常動作を確保せしめ処理性能を向上させた情報処理装置を実現することを目的とする。

【0015】

【課題を解決するための手段】

この課題を解決するために本発明の情報処理装置は、パイプラインで制御される情報処理装置において、パイプラインが、論理演算器におけるクリティカルパスを正常に実行するためのクロック周波数より高速なクロック周波数で駆動され、パイプラインの実行ステージにおいて、前記パイプラインのクロック周波数で駆動する高速駆動の論理演算器と、前記クリティカルパスを正常に実行するためのクロック周波数以下のクロック周波数で駆動する低速駆動の論理演算器を備え、前記高速駆動の論理演算器が正しく論理演算処理が実行できた場合は、当該論

理演算結果をパイプライン実行ステージの出力とし、前記高速駆動の論理演算器が正しく論理演算処理が実行できなかった場合は、前記低速駆動の論理演算器の論理演算結果をパイプライン実行ステージの出力とすることを特徴とする。

【 0 0 1 6 】

上記構成により、高速駆動の論理演算器と低速駆動の論理演算器の2種類の論理演算器を用い、高速駆動の論理演算器による高速処理により情報処理装置全体の処理速度向上を図ることができ、かつ、高速駆動の論理演算器では処理が間に合わないクリティカルパスのデータに対しては低速駆動の論理演算器により確実に処理を実行し、情報処理装置全体の正常処理を確保せしめることができる。

【 0 0 1 7 】

ここで、前記低速駆動の論理演算器が、複数の低速駆動の論理演算器からなり、実行ステージで用いる低速駆動の論理演算器を実行ステージごとに順番に切り替え、各低速駆動の論理演算器は、自分が担当した実行ステージの演算を前記クリティカルパスを正常に実行するためのクロック周波数以下のクロック周波数で処理することが好ましい。

【 0 0 1 8 】

上記構成により、複数の低速駆動の論理演算器により、高速なクロック周波数で駆動されているパイプラインの実行ステージを1つずつ順番に分担して担当し、たとえ自分が担当した実行ステージの演算がクリティカルパスであっても、それを十分処理できる低速駆動により演算を実行し、確実に正しい演算結果を得ることができ、パイプラインの駆動速度で高速駆動している論理演算器の実行結果が正しいものか参照することができる。

【 0 0 1 9 】

ここで、前記パイプラインのクロック周波数が、前記クリティカルパスを正常に実行するためのクロック周波数の n 倍である場合、前記複数の低速駆動の論理演算器を n 個の低速駆動の論理演算器とし、パイプラインにおいて順に実行される n 個の実行ステージの演算を、 n 個の低速駆動の論理演算器が順に担当して行くことが好ましい。

【 0 0 2 0 】

つまり、クリティカルパスを正常に演算実行する低速駆動の n 倍の周波数で駆動するパイプラインのすべての実行ステージを分担するために、 n 個の低速駆動の論理演算器を用意し、各実行ステージを担当させるわけである。

【0021】

次に、本発明の情報処理装置において、同じ実行ステージの演算結果として得られた前記高速駆動の論理演算器の出力と前記低速駆動の論理演算器の出力とを入力として両者を比較する比較器を備え、パイプライン実行ステージの出力処理にあたり、前記高速駆動の論理演算器の出力をパイプライン実行ステージの出力として仮定しておき、前記比較器における比較処理結果が一致を示す場合は、前記高速駆動の論理演算器の出力を実行ステージの出力として確定してパイプライン処理を継続し、前記比較器における比較処理結果が一致を示さない場合は、前記低速駆動の論理演算器の出力を実行ステージの出力として置き換えることが好ましい。

【0022】

上記構成により、パイプラインの実行ステージの通常の出力として前記高速駆動の論理演算器の出力をそのまま用いて通常のパイプライン処理を高速に実行し、かつ、演算処理結果が正しいものであるか否かを低速駆動の論理演算器の出力との比較により確認し、クリティカルパスなど正しい演算結果が得られていないものが検知されれば、低速駆動の論理演算器の出力で置き換えるという一種のエラー処理を行うことにより情報処理装置の正常動作を確保せしめることができる。

【0023】

ここで、前記比較器における比較処理結果が一致を示さない場合において、前記低速駆動の論理演算器の出力を実行ステージの出力として置き換える処理が完了するまで、パイプラインの全ステージの駆動を一時停止することが好ましい。

【0024】

上記のエラー処理として、実行ステージの出力を低速駆動の論理演算器の出力とする置き換え処理のために費やされるマシンサイクル分、パイプラインを一時停止し、パイプライン各ステージの同期を採るためである。

【 0 0 2 5 】

次に、本発明の情報処理装置において、一定期間の前記不一致検出信号の発生回数をカウントする回路と、前記カウント数によってパイプラインを駆動する周波数を変化させる回路を備えることが好ましい。

【 0 0 2 6 】

上記構成により、情報処理装置が実際に処理するデータにおいて、クリティカルパスの発生頻度を調べることができ、クリティカルパスの発生頻度に応じてパイプラインを駆動するクロック周波数を変化させることができる。つまり、クリティカルパスが発生すると低速駆動の論理演算器の出力へ置き換えるというエラー処理が発生するので、クリティカルパス発生頻度が所定より高い場合にはエラー処理による処理効率低下が大きくなり、むしろパイプライン周波数を下げた方が全体として有利な場合もあるからである。

【 0 0 2 7 】

ここで、前記パイプラインのクロック周波数を高めることによる前記高速駆動の論理演算器の処理量の増加と、前記パイプラインのクロック周波数を高めることにより前記高速駆動の論理演算器が正しく論理演算処理ができず、前記低速駆動の論理演算器の論理演算結果をパイプライン実行ステージの出力とする処理が増加することによる処理遅延量の増加とを比較し、前者が後者に対して所定割合より大きい場合にパイプラインのクロック周波数を高めることが好ましい。

【 0 0 2 8 】

また、前記パイプラインのクロック周波数を下げることによる前記高速駆動の論理演算器の処理量の低下と、前記パイプラインのクロック周波数を下げることにより前記高速駆動の論理演算器が正しく論理演算処理ができ、前記低速駆動の論理演算器の論理演算結果をパイプライン実行ステージの出力とする処理が減少することによる処理遅延量の低下とを比較し、後者が前者に対して所定割合より大きい場合にパイプラインのクロック周波数を下げることが好ましい。

【 0 0 2 9 】

つまり、パイプラインの周波数を高めることによる前記高速駆動の論理演算器の処理量増加のメリットと、クリティカルパス発生による低速駆動の論理演算器

の出力への置き換えというエラー処理発生というデメリットを比較考量し、メリットの方が大きい場合にはパイプラインの周波数を高めることによりスループット向上を図ることができる。逆に、パイプラインの周波数を下げることによる前記高速駆動の論理演算器の処理量低下のデメリットと、クリティカルパス発生の減少による低速駆動の論理演算器の出力への置き換えというエラー処理が減少するメリットを比較考量し、メリットの方が大きい場合にはパイプラインの周波数を下げることによりスループット向上を図ることができる。

【 0 0 3 0 】

次に、本発明の情報処理装置において、複数の論理演算器と、クリティカルパスとなるテストデータを生成するデータ生成回路と、各論理演算器のクリティカルパス実行時間を測定する実行時間測定回路と、クリティカルパス実行時間が最短となる論理演算器を検出する検出回路を備え、前記検出回路によって検出された論理演算器を前記高速駆動の論理演算器とし、他の一または複数の論理演算器を前記低速駆動の論理演算器として選択することが好ましい。

【 0 0 3 1 】

上記構成により、情報処理装置において用意された複数の論理演算器のうち、クリティカルパスとなるテストデータを最も高速に実行できる論理演算器を検知して高速駆動の論理演算器として用いることにより、情報処理装置の処理効率の最大化を図ることができる。ここで、各論理演算器のクリティカルパス実行時間とは、各論理演算器ユニット内部でのクリティカルパス演算処理時間としても良く、また、各論理演算器ユニット内部での演算処理時間および各論理演算器ユニットからの出力結果を格納するレジスタに論理演算結果が格納されるまでの時間の和として評価しても良い。

【 0 0 3 2 】

【発明の実施の形態】

以下、本発明の実施の形態について、図 1 から図 9 を用いて説明する。

【 0 0 3 3 】

(実施形態 1)

図 1 は、本発明の第 1 の実施形態における情報処理装置のブロック図である。

【 0 0 3 4 】

同図において、I F B 1、命令レジスタ 2、命令レジスタ 2 の出力 3、D E C B 4、デコード信号レジスタ 5、デコード信号レジスタ 5 の出力 6 は、従来技術で説明した各要素と同様である。

【 0 0 3 5 】

本実施形態 1 では、E X B 7 には、パイプラインのクロック周波数と同じクロック周波数で駆動する高速駆動の A L U 2 0 と、クリティカルパスの演算が実行可能なクロック周波数で駆動する低速駆動の A L U 2 1 および A L U 2 2 の 2 種類の A L U を含んでおり、低速駆動の A L U としては A L U 2 1 と A L U 2 2 の 2 つが含まれている。A L U 2 1 には入力データ A L U A、A L U B を保持するラッチ 8、9 と制御信号 A L U C N T を保持するレジスタ 1 0 が接続され、A L U 2 2 には入力データ A L U A、A L U B を保持するラッチ 1 1、1 2 と制御信号 A L U C N T を保持するラッチ 1 3 が接続されている。

【 0 0 3 6 】

さらに、A L U 2 1 の出力と A L U 2 2 の出力が入力されたセクタ 2 6 と、A L U 2 0 の出力 2 3 とセクタ 2 6 の出力が入力されたセクタ 2 8 を備え、さらにそのセクタ 2 8 の出力が入力されたレジスタ 3 0 と、前記レジスタ 3 0 の出力とセクタ 2 6 の出力が入力された比較器 3 2 を備え、前記比較器 3 2 より入力値の一致を示す信号 3 3 と入力値の不一致を示す信号 3 4 を出力し、不一致信号 3 4 は I F B 1 と D E C B 4 へ入力されている。

【 0 0 3 7 】

また、一致信号 3 3 とデコードレジスタが有効になる最初のサイクルを示す D R V F S T によって、A L U 2 1 への入力データ、及び制御信号のラッチ 8、9、1 0 のイネーブル信号 3 8 と、A L U 2 2 への入力データ、及び制御信号のラッチ 1 1、1 2、1 3 のイネーブル信号 3 9 が生成されている。

【 0 0 3 8 】

ここで、セクタ 2 6 が設けられているのは、パイプラインの各実行ステージで用いる低速駆動の A L U 2 1 と A L U 2 2 を実行ステージごとに交互に切り替えるためである。図 1 の構成では、各低速駆動の A L U 2 1 および A L U 2 2 は

、実行ステージを交互に担当し、それぞれの演算を、低いクロック周波数、つまり、パイプライン周波数の2分の1のクロック周波数で実行することができる。このように各低速駆動のALUは、クリティカルパスを正常に実行するためのクロック周波数以下のクロック周波数で演算を実行することができ、クリティカルパスの正しい演算結果を得ることが可能となる。

【0039】

なお、図1の例では、低速駆動のALU21とALU22の2つとしたが、パイプラインの周波数が、クリティカルパスを正常に実行するためのクロック周波数のn倍である場合、n個の低速駆動のALUを用意し、パイプラインの連続するn個の実行ステージのそれぞれの演算を、n個の各低速駆動のALUそれぞれが順番に担当するように切り替える。各低速駆動のALUはn個あるので、1つあたりパイプラインのクロック周波数のn分の1のクロック周波数で実行することができる。

【0040】

次に、比較器32が設けられているのは、同じ実行ステージの論理演算結果として得られた高速駆動のALU20の出力と低速駆動のALU21またはALU22の出力とを比較するためである。図1の構成では、パイプラインの実行ステージの出力として投機的にまず選ばれる高速駆動のALU20の検算結果出力を格納した出力レジスタ30の出力と、比較すべき同じ実行ステージを担当した低速駆動のALUが選択されたセレクタ26の出力が比較される。

【0041】

比較器32における比較処理結果が一致を示す場合は、高速駆動のALU20が正しく演算処理を実行しているのでパイプラインを停止させることなく、出力レジスタ30の出力を実行ステージの出力とする。一方、比較器32における比較処理結果が一致を示さない場合は、クリティカルパスなど高速駆動のALU20が正しく演算処理を実行できていないので、比較器32の信号34によりセレクタ28を切り替えて低速駆動のALU21またはALU22の出力を選択して出力レジスタ30のデータを置き換え、実行ステージの出力とする。このような一種のエラー処理を行う。

【 0 0 4 2 】

また、比較器 3 2 における比較処理結果が一致を示さない場合は、低速駆動の A L U 2 1 または A L U 2 2 の出力を出力レジスタ 3 0 に置き換える処理が完了するまで、パイプラインの全ステージの駆動を一時停止することによりパイプライン駆動の同期を維持する。比較器 3 2 の不一致信号 3 4 を I F B 1 および D E C B 4 に入力して各ステージを停止させ、さらに、一致信号 3 3 が出力されず A L U 入力イネーブル信号 3 8、3 9 が生成されないため、低速駆動の A L U 2 1、A L U 2 2 の入力段が停止され、実行ステージでの同期も維持される。

【 0 0 4 3 】

以上のように構成された本実施形態の情報処理装置の動作について図 2 のタイミングチャートを用いて詳しく説明する。

【 0 0 4 4 】

図中、a の期間では命令 1 のフェッチが I F B 1 で行われる。

【 0 0 4 5 】

次に、b の期間の開始で命令 1 の命令コードが命令レジスタ 2 へ格納され、さらに D E C B 4 においてデコードされ、命令 1 のデコード信号が生成される。また、I F B 1 で次の命令 2 のフェッチが行われる。

【 0 0 4 6 】

次に、c の期間の開始で命令 1 のデコード信号がデコード信号レジスタ 5 へ格納され、E X B 7 においてデコード信号レジスタ 5 から出力される命令 1 のデコード信号によって対象レジスタ間の演算が行われる。

【 0 0 4 7 】

まず、パイプラインの実行ステージとして高速駆動の A L U 2 0 において命令 1 が実行され、A L U 2 0 の出力 2 3 として命令 1 の実行結果が得られる。一方、低速駆動の A L U の選択が行われる。c の区間からデコードレジスタバリッド信号 D R C R V が有効になり、D E C R V の最初のサイクルを示す信号 D R V F S T が有効になる。c の区間においてイネーブル信号 3 8 はハイになり、レジスタ 8、9、1 0 には命令 1 を実行するためのデータ、制御信号が入力される。つまり、c の期間において A L U 2 1 が選択され、A L U 2 1 は 2 マインサイクル

をかけて演算を実行し、その出力 2 4 が現れる。

【 0 0 4 8 】

また、c の期間では、I F B 1 においては命令 3 のフェッチが行われ、D E C B 4 においては命令 2 のデコードが行われる。

【 0 0 4 9 】

次に、d の区間では、以下のように E X レジスタ 3 0 へ命令 1 によって行われた演算結果が格納され、E X B 7 においては命令 2 の演算が行われる。

【 0 0 5 0 】

まず、パイプラインの実行ステージとして高速駆動の A L U 2 0 において命令 2 が実行され、A L U 2 0 の出力 2 3 として命令 2 の実行結果が得られる。一方、命令 2 を実行する低速駆動の A L U は c の期間において選択されていた A L U 2 1 から A L U 2 2 に切り換えられる。セレクト 2 8 にはセレクト 2 6 の出力と A L U 2 0 の出力が入力され、不一致信号 3 4 がハイの場合のみセレクト 2 6 の出力を選択し、レジスタ 3 0 へ入力する。イネーブル信号 3 8 は D R V F S T を受けてローレベルへ変化し、レジスタ 8、9、10 には命令 1 を実行するためのデータ、制御信号が保持される。一方、d の区間ではイネーブル信号 3 9 がハイレベルとなり、レジスタ 11、12、13 には命令 2 を実行するためのデータ、制御信号が入力される。このように、A L U 2 1 は d の期間も c の期間と同様に命令 1 の実行を行い、結局 c、d の 2 サイクルの期間をかけて命令 1 の実行が行われる。A L U 2 2 は d の期間から命令 2 の実行を開始し、出力 2 5 として命令 2 の実行結果が現れる。なお、後述するように A L U 2 2 は e の期間も命令 2 の実行を行うので、結局 d、e の 2 サイクルの期間をかけて命令 2 の実行を行うこととなる。

【 0 0 5 1 】

また、d の期間においては命令 1 の実行結果の確認・出力処理がなされる。まず、E X レジスタ 3 0 の有効を示す E X R V が H I となり、比較器 3 2 へ入力される。E X R V がハイでかつ E X レジスタ 3 0 とセレクト 2 6 で選択された低速駆動の A L U の出力が一致している場合は一致信号 3 3 がハイレベルになり、不一致の場合は不一致信号 3 4 がハイレベルになる。d の期間ではイネーブル信号

39がハイレベルなのでALU21の出力がセクタ26によって選択され、ここでは比較器32においてALU21の出力とレジスタ30の出力が比較される。つまり、高速駆動ALU20が1マシンサイクルで実行した命令1の実行結果と低速駆動ALU21が2マシンサイクルかけて実行した命令1の実行結果とが比較される。この例では命令1は基本サイクル内に完了していたため、レジスタ30の出力とALU21の出力が一致しているので一致信号33がハイとなり、レジスタ30の出力（つまり、高速駆動ALU20が1マシンサイクルで実行した命令1の実行結果）がパイプライン実行ステージにおけるEXB7の出力となる。このように、命令1は高周波数により正常にパイプライン実行されたことが分かる。

【0052】

また、dの期間中、IFB1においては命令4のフェッチを行い、DECB4においては命令3のデコードが行われる。

【0053】

次に、高速駆動ALU20が1マシンサイクルで実行できない場合の処理動作をe、f、gの3区間かけて説明する。eの区間では高速駆動ALU20が1マシンサイクルで命令3の実行を行っているが、この例では命令3がクリティカルパスであり、高速駆動ALU20が命令3を1マシンサイクルで実行完了できない演算となっているものとする。

【0054】

eの区間では以下のように、EXレジスタ30へ命令2によって行われた演算結果が格納され、EXB7においては命令3の演算が行われる。なお、cの期間と同様の部分の説明は簡単に記述した。

【0055】

まず、パイプラインの実行ステージとして高速駆動のALU20において命令3が実行され、ALU20の出力23として命令3の実行結果“3”が現れる。

【0056】

一方、命令3を実行する低速駆動のALUはdの期間において選択されていたALU22からALU21に切り換えられる。eの区間では前サイクルの一致信

号 3 3 の H I を受けてイネーブル信号 3 8 が反転し、ハイとなる。それに従ってレジスタ 8、9、10 には命令 3 を実行するためのデータと制御信号が入力される。イネーブル信号 3 9 は e の期間中ローのためレジスタ 11、12、13 には命令 2 を実行するためのデータと制御信号が保持される。このように、ALU 22 は e の期間も d の期間と同様に命令 2 の実行を行い、結局 d、e の 2 サイクルの期間をかけて命令 2 の実行が行われる。ALU 21 は e の期間から 2 マシンサイクルかけて命令 3 の実行を開始し、最初の 1 マシンサイクル時点での出力 25 として命令 3 の実行結果 “3” が現れる。なお、後述するように ALU 21 は 2 マシンサイクル目の次の f の期間において正しい実行結果 “3' ” を得ることとなる。

【 0 0 5 7 】

なお、e の期間における命令 2 の実行結果の確認・出力処理としては、イネーブル信号 3 9 がローなので ALU 22 の出力がセクタ 26 によって選択され比較器 32 において ALU 22 の出力とレジスタ 30 の出力が比較される。命令 2 も基本サイクル内に完了しているため一致信号 33 がハイとなり、e の期間においてもレジスタ 30 の出力、つまり、高速駆動 ALU 20 が 1 マシンサイクルで実行した命令 2 の実行結果がパイプライン実行ステージにおける EXB 7 の出力となる。

【 0 0 5 8 】

次に、f の期間において、以下のように、EX レジスタ 30 に格納された高速駆動の ALU 20 の命令 3 の演算結果が、低速駆動の ALU 21 の命令 3 の演算結果に置き換えられる。

【 0 0 5 9 】

まず、高速駆動 ALU 20 は、デコードレジスタ 5 から次の命令 4 を受け取り、実行を開始する。ALU 20 の出力 23 として命令 4 の実行結果 “4” が現れる。命令 4 を実行する低速駆動の ALU は d の期間と同様の動作により、e の期間において選択されていた ALU 21 から ALU 22 に切り換えられる。

【 0 0 6 0 】

一方、e、f の 2 マシンサイクルをかけてクリティカルパスである命令 3 を実

行していたALU21は、2マシンサイクル目のfの期間で正しい演算結果を得たものとする。つまり、当初、高速駆動のALU20と同じく“3”と得られていた信号が正しい信号“3'”に変わっている。

【0061】

次に、fの期間における命令3の実行結果の確認・出力処理がなされる。fの区間ではセクタ26においてALU21の出力24とレジスタ30の出力31を比較しているがALU21の出力24の値が3から3'へ変化した時点で不一致信号34がハイになり、一致信号33はローになる。なお、セクタ28には不一致信号34が制御信号として与えられるので、次のg期間においてセクタ28はセクタ26の出力27、つまり、低速駆動ALU21の出力24である“3'”を選択する。

【0062】

一方、上記置き換え処理発生のため、パイプラインステージ、各論理回路の動作を一時停止させ、状態をgの区間まで維持しなければならない。不一致信号34はIFB1とDECB4へ入力され、IFB1においては次のサイクルgにかけて命令レジスタ2の保持と命令6の再フェッチを行い、DECB4においてはデコードレジスタ5の保持が行われる。

【0063】

gの期間ではイネーブル信号38、39の反転は行われず、レジスタ8、9、10は保持される。前サイクルの不一致信号34のハイを受けてセクタ28でセクタ26の出力が選択され、レジスタ30へセクタ26の出力値“3'”が格納される。この置き換えによりレジスタ30の値とセクタ出力27の値が一致するため一致信号33がハイレベルへ変化する。

【0064】

hの期間においては前サイクルの一致信号33のハイを受けてイネーブル信号38、39が反転し、通常の動作を繰り返すこととなる。

【0065】

以上より、本実施形態1においてはパイプラインの通常の実行ステージで用いられる高速駆動のALU20の他に、2つの低速駆動ALU（ALU21、AL

U 2 2) を備え、かつ、前記別 A L U 2 1、2 2 の入力データのそれぞれのイネーブル信号 3 8、3 9 が常に反対の論理を示し、A L U 2 0 への入力データが更新される際にはイネーブル信号が反転され、A L U 2 1、2 2 のうちの必ずどちらかの A L U の入力データが保持されることにより、パイプラインで実行される全ての命令を途絶えることなく交互に 2 サイクルかけて実行することが可能となっている。また、高速駆動の A L U 2 0 と他の 2 つの低速駆動の A L U 2 1、2 2 のうちの比較対象となっている A L U の実行結果に不一致が発生した場合、つまり、クリティカルパスによる高速駆動の A L U 2 0 の実行が基本サイクル内で正常に終了しない場合には、実行結果レジスタ 3 0 へ比較対象の低速駆動の A L U の実行結果を格納し直すことにより置き換え、高速なパイプライン駆動によるエラー発生を手当てして処理することを可能としている。なお、上記置き換え時は、命令レジスタ 2 とデコード信号レジスタ 5 の値を保持することにより、その後の命令実行の不具合が発生することなく処理を継続することを可能としている。

【 0 0 6 6 】

なお、以上の例では、低速駆動の A L U としては A L U 2 1 と A L U 2 2 の 2 つを用いた例としたが、上記の通り、n 個の低速駆動の A L U を用い、パイプラインの駆動周波数をクリティカルパス実行に必要なクロック周波数の n 倍のクロック周波数にまで高速化させることが可能である。

【 0 0 6 7 】

このように、本実施形態 1 の情報処理装置は、高速駆動の論理演算器と低速駆動の論理演算器の 2 種類の論理演算器を用い、高速駆動の論理演算器による高速処理により情報処理装置全体の処理速度向上を図ることができ、かつ、高速駆動の論理演算器では処理が間に合わないクリティカルパスのデータに対しては低速駆動の論理演算器により確実に処理を実行し、情報処理装置全体の正常処理を確保せしめることができる。

【 0 0 6 8 】

(実施形態 2)

次に、実施形態 2 にかかる情報処理装置を説明する。実施形態 2 の情報処理装

置は、実際に処理するデータに含まれるクリティカルパスの発生割合を考慮してパイプライン駆動周波数を調整する機能を備えたものである。

【 0 0 6 9 】

本実施形態 2 では、実際に処理するデータに含まれるクリティカルパスの発生割合を検出するため一定期間の前記不一致検出信号の発生回数をカウントする回路を備え、また、カウント数によってパイプラインを駆動する周波数を変化させる回路を備えた構成とする。クリティカルパス発生割合に応じたクロック周波数の調整は、クリティカルパス発生回数の上限下限のしきい値を設けておき、実施形態 1 で説明した比較器 3 2 の不一致信号 3 4 の所定期間あたりの発生回数が上限しきい値を超えたときはパイプライン駆動周波数を下げ、下限しきい値を下回ったときはパイプライン駆動周波数を上げるという方法で行う。

【 0 0 7 0 】

図 3 は、本発明の第 2 の実施形態にかかる、上記第 1 の調整方法を行う情報処理装置のブロック図である。同図の情報処理装置においては、比較器 3 2 の不一致信号 3 4 がオーバーフロー検出器 4 0 に入力されている。オーバーフロー検出器 4 0 は検出期間設定レジスタ 4 1 と最小検出回数設定レジスタ 4 2 と最大検出回数設定レジスタ 4 3 を備えている。クロック制御回路 4 6 は初期周波数設定レジスタ 4 7 と周波数変化量設定レジスタ 4 8 を備えている。オーバーフロー検出器 4 0 からはクロック周波数の高速化を要求する信号 CLKUP 4 4 とクロック周波数の低速化を要求する信号 CLKDOWN 4 5 がクロック制御回路 4 6 に入力されている。

【 0 0 7 1 】

パイプライン駆動周波数は、まず初期周波数設定レジスタ 4 7 に設定された周波数で駆動され、その後、クロック制御回路 4 6 により周波数変化量設定レジスタ 4 8 に設定された値きざみで調整されていく。もし、オーバーフロー検出器 4 0 が検出したオーバーフロー回数が最大検出回数設定レジスタ 4 3 の値を超えた場合は CLKDOWN 4 5 がハイになり、クロック制御回路 4 6 により現状の駆動周波数より周波数変化量設定レジスタ 4 8 に設定された値だけ低速化されたクロック CLK が次のサイクルより出力される。もし、検出期間設定レジスタ 4 1

に設定された期間に不一致信号 3 4 が発生した回数が最小検出回数設定レジスタ 4 2 の値を下回った場合は CLKUP 4 4 がハイになり、クロック制御回路 4 6 により、現状の周波数より周波数変化量設定レジスタ 4 8 に設定された値だけ高速化されたクロック CLK が次のサイクルより出力される。

【 0 0 7 2 】

この構成により不一致の発生が多発する場合に CLK を低速にして、システム性能の低下を抑え、不一致が未発生の場合に CLK を高速にして制御装置の性能を向上させることを可能としている。

【 0 0 7 3 】

ここで、上限しきい値と下限しきい値を如何に設定するかは重要である。パイプライン駆動周波数の調整は、結局、パイプライン駆動周波数を高めることによる処理量増加と、パイプライン駆動周波数を高めるにつれ増加するクリティカルパス発生に対するエラー処理による処理遅延との両者を比較考量し、前者が大き過ぎる場合にはパイプライン駆動周波数を上げ、後者が大き過ぎる場合にはパイプライン駆動周波数を下げることを狙いとして実行するので、この関係を満たすようにオーバーフロー回数の上限しきい値と下限しきい値を設定する。

【 0 0 7 4 】

以下、オーバーフロー回数の上限しきい値と下限しきい値の決め方について説明する。図 4 は、各命令の実行時間の分布の一例であり、ここでは説明の便宜上、各命令の実行時間の分布はほぼ正規分布に従うと仮定した。図 4 に見るように、この例ではおよそ 7 0 % の命令が 1 0 ナノ秒以内でその処理を完了し、およそ 9 0 % の命令が 1 1 ナノ秒以内でその処理を完了し、およそ 9 7 % の命令が 1 2 ナノ秒以内でその処理を完了していることが分かる。図 4 の分布を示す命令実行において、パイプラインの駆動周波数が 1 0 ナノ秒と 1 1 ナノ秒と 1 2 ナノ秒のそれぞれである場合に、1 0 0 個の命令を実行するために必要な時間を計算して見る。なお、エラー処理にはペナルティとして基本サイクル分のマシンサイクル（ここでは 1 マシンサイクル）が費やされるとする。

【 0 0 7 5 】

（1 0 ナノ秒のときの処理時間） $70 \times 10 + 30 \times 20 = 1300$

(11ナノ秒のときの処理時間) $90 \times 11 + 10 \times 22 = 1210$

(12ナノ秒のときの処理時間) $97 \times 12 + 3 \times 24 = 1236$

つまり、図4のような分布を示す命令群に対しては、1サイクル11ナノ秒になるように、つまり、オーバーフローが命令全体のおよそ10%程度となるようにパイプライン駆動周波数を調整することが好ましいと言える。上記例ではオーバーフロー回数の上限しきい値は命令全体のおよそ30%の回数(10ナノ秒の場合に相当する回数)より小さいものとし、下限しきい値は命令全体のおよそ3%の回数(12ナノ秒の場合に相当する回数)より大きいものとして定めれば良い。例えば、100命令に対して、上限しきい値を29回、下限しきい値を4回などとする。

【0076】

以上のように命令群の処理実行時間の分布が既知であれば上限しきい値である最大検出回数設定レジスタ43の値と下限しきい値である最小検出回数設定レジスタ42の値を設定することができる。

【0077】

次に、汎用的に利用される場合など実際に処理する命令群の分布が特定できない場合の調整方法について検討する。例えば、初期マシンサイクル設定レジスタ47の値を t ナノ秒とし、マシンサイクル変化量設定レジスタ48の値を Δt ナノ秒とし、検出期間設定レジスタ41の値を m とし、命令のうち t ナノ秒におけるオーバーフロー命令数を n とし、 Δt ナノ秒分マシンサイクルを変化させたときのオーバーフロー命令変化数を Δn とする。

【0078】

ここで、オーバーフロー回数下限しきい値(以下において1マシンサイクル時間を増加させ、速すぎるパイプライン駆動周波数を下げて調整する方向のしきい値とする)と、オーバーフロー回数上限しきい値(以下において1マシンサイクル時間を減少させ、余裕のあるパイプライン駆動周波数を上げて調整する方向のしきい値とする)を検討する。

【0079】

初期マシンサイクルでの処理時間は、(数1)となる。

【0080】

【数1】

$$(t \times m + t \times n)$$

まず、オーバーフロー回数下限しきい値を検討する。1マシンサイクルを Δt ナノ秒分増加させたときの処理時間は、(数2)となる。

【0081】

【数2】

$$(t + \Delta t) \times m + (t + \Delta t) (n - \Delta n)$$

(数2)が(数1)より小さい場合、マシンサイクル時間を増やした方が全体のスループットが向上することを意味する。

【0082】

いま、 $t = 10$ ナノ秒、 $\Delta t = 1$ ナノ秒、 $m = 100$ とすると、(数3)が導ける。

【0083】

【数3】

$$\Delta n > (100 + n) / 11$$

(数3)をグラフ化したものが図5(a)である。図5(a)の関係を満たすようにオーバーフロー回数下限値を設定しておき、パイプライン駆動周波数を1ナノ秒増加させた場合に当該オーバーフロー下限値を下回る場合にはパイプラインマシンサイクルを1ナノ秒増加させる調整を行えば良い。

【0084】

次に、オーバーフロー回数上限しきい値を検討する。1マシンサイクルを Δt ナノ秒分減少させたときの処理時間は(数4)となる。

【0085】

【数4】

$$(t - \Delta t) \times m + (t - \Delta t) (n + \Delta n)$$

(数4)が(数1)より小さい場合、マシンサイクル時間を減らした方が全体のスループットが向上することを意味する。

【0086】

いま、 $t = 10$ ナノ秒、 $\Delta t = 1$ ナノ秒、 $m = 100$ とすると、(数5) が導ける。

【0087】

【数5】

$$\Delta n < (100 + n) / 9$$

(数5) をグラフ化したものが図5 (b) である。図5 (b) の関係を満たすようにオーバーフロー回数上限値を設定しておき、パイプラインマシンサイクルを1ナノ秒減少させた場合に当該オーバーフロー上限値を上回る場合にはパイプラインマシンサイクルを1ナノ秒減少させる調整を行なう必要はない。

【0088】

つまり、図5はパイプラインの初期駆動周波数を1マシンサイクル10ナノ秒相当とした場合、1マシンサイクルを1ナノ秒分増加または減少させた場合、オーバーフロー回数がどれほど変化すれば全体として処理時間が短縮できるかという境界を示す図となっている。例えば、パイプライン駆動周波数が1マシンサイクル10ナノ秒相当の場合にオーバーフロー回数が30回の場合、1マシンサイクルを1ナノ秒分増加させた場合にオーバーフロー回数が12回以上減少し、18回以下である場合は、パイプラインマシンサイクルを11ナノ秒に落とした方が有利になると判断できる。ちなみに上記例では10ナノ秒相当の場合の処理時間が1300ナノ秒に対して11ナノ秒相当の場合にオーバーフロー回数が18回の場合の処理時間が1298ナノ秒である。

【0089】

また、同様に、パイプライン駆動周波数が1マシンサイクル10ナノ秒相当の場合にオーバーフロー回数が30回の場合、1マシンサイクルを1ナノ秒分減少させた場合にオーバーフロー回数が14回以下しか増加せず、44回以下であれば、パイプラインマシンサイクルを9ナノ秒に高めた方が有利になると判断できる。ちなみに10ナノ秒相当の場合の処理時間が1300ナノ秒に対して9ナノ秒相当の場合にオーバーフロー回数が44回の場合の処理時間が1296ナノ秒である。

【0090】

以上、一定期間の前記不一致検出信号の発生回数をカウントする回路と、前記カウント数によってパイプラインを駆動する周波数を変化させる回路を備えることにより、パイプライン駆動周波数の調整を加味して情報処理装置のスループットの向上を図ることができる。

【 0 0 9 1 】

(実施形態 3)

次に、実施形態 3 にかかる情報処理装置を説明する。実施形態 3 の情報処理装置は、複数の論理演算器と、クリティカルパスとなるテストデータを生成するデータ生成回路と、各論理演算器のクリティカルパス実行時間を測定する実行時間測定回路と、クリティカルパス実行時間が最短となる論理演算器を検出する検出回路を備え、検出回路によって検出された論理演算器を高速駆動の論理演算器とし、他の一または複数の論理演算器を低速駆動の論理演算器として選択するものである。

【 0 0 9 2 】

図 6 は、本発明の第 3 の実施形態における情報処理装置ブロック図である。

【 0 0 9 3 】

同図の情報処理装置においては、E X B 7 内にクリティカルパスとなるテストデータを生成するクリティカルデータ生成装置 4 9 を備えている。クリティカルデータ生成装置 4 9 からは各 A L U それぞれに対して入力するクリティカルデータ 5 0、5 1 と、クリティカルとなる演算を実行するための制御信号 5 2 とクリティカルセレクト信号 C D S E L 5 3 が出力されている。

【 0 0 9 4 】

A L U 2 0 のデータ入力には A L U 入力データラッチ 6 0、6 1 が接続され、制御信号入力には A L U 制御信号ラッチ 6 2 が接続されている。A L U 2 1 のデータ入力には A L U 入力データラッチ 8、9 が接続され、制御信号入力には A L U 制御信号ラッチ 1 0 が接続されている。A L U 2 2 のデータ入力には A L U 入力データラッチ 1 1、1 2 が接続され、制御信号入力には A L U 制御信号ラッチ 1 3 が接続されている。

【 0 0 9 5 】

ALUAとクリティカルデータ50はセクタ54へ入力され、セクタ54の出力57はALU入力データラッチ60、8、11へ入力されている。ALUBとクリティカルデータ51はセクタ55へ入力され、セクタ55の出力58はALU入力データラッチ61、9、12へ入力されている。ALUCNTとクリティカル制御信号52はセクタ56へ入力され、セクタ56の出力59はALU制御信号ラッチ62、10、13へ入力されている。

【0096】

ALU20、ALU21、ALU22の各出力は実行時間測定器63へ入力され、各ALUにおけるテストデータ実行時間が測定される。実行時間測定器63からはパイプラインALU選択信号PALUSEL(3:1)とALU20の入力データラッチおよび制御信号ラッチのイネーブル信号64とALU21の入力データラッチおよび制御信号ラッチのイネーブル信号65とALU22の入力データラッチおよび制御信号ラッチのイネーブル信号66が生成されている。

【0097】

図7は、実行時間測定器63の詳しい回路図である。

【0098】

図中において、ALU20の出力の全ビットが入力されたNORゲート67の出力が遅延バッファ69とANDゲート75とSRFF68のSET端子に入力されている。遅延バッファ69の出力は遅延バッファ70とANDゲート74とに入力されていて、遅延バッファ70の出力は遅延バッファ71とANDゲート73とに入力されていて、遅延バッファ71の出力はANDゲート72に入力されている。ANDゲート72の出力はSRFF76のSET端子に、ANDゲート73の出力はSRFF77のSET端子に、ANDゲート74の出力はSRFF78のSET端子に、ANDゲート75の出力はSRFF79のSET端子にそれぞれ入力されている。

【0099】

ALU21の出力の全ビットが入力されたNORゲート80の出力は遅延バッファ82とANDゲート88とSRFF81のSET端子に入力されている。遅延バッファ82の出力は遅延バッファ83とANDゲート87とに入力されてい

て、遅延バッファ83の出力は遅延バッファ84とANDゲート86とに入力されていて、遅延バッファ84の出力はANDゲート85に入力されている。ANDゲート85の出力はSRFF89のSET端子に、ANDゲート86の出力はSRFF90のSET端子に、ANDゲート87の出力はSRFF91のSET端子に、ANDゲート88の出力はSRFF92のSET端子にそれぞれ入力されている。

【0100】

ALU22の出力の全ビットが入力されたNORゲート93の出力は遅延バッファ95とANDゲート101とSRFF94のSET端子に入力されている。遅延バッファ95の出力は遅延バッファ96とANDゲート100とに入力されていて、遅延バッファ96の出力は遅延バッファ97とANDゲート99とに入力されていて、遅延バッファ97の出力はANDゲート98に入力されている。ANDゲート98の出力はSRFF102のSET端子に、ANDゲート99の出力はSRFF103のSET端子に、ANDゲート100の出力はSRFF104のSET端子に、ANDゲート101の出力はSRFF105のSET端子にそれぞれ入力されている。

【0101】

FSFF76、77、78、79、89、90、91、92、102、103、104、105の出力をデコードして、PALUSEL(3:1)を生成している。

【0102】

以上のように構成された本実施形態の情報処理装置の動作について図8、図9のタイミングチャートを用いて説明する。

【0103】

図8中a-4のタイミングからCDSEL53が4サイクルの期間有効となる。CDSEL53がHIの区間はクリティカルデータ生成器49からの出力50、51、52がセレクタ54、55、56において選択される。ALU20、21、22の各入力データのイネーブル信号64、65、66もa-3から3サイクルの期間有効となる。a-3とa-2の2サイクルの間に各ALUの実行時間

を測定し、 $a-1$ のサイクルからパイプラインALU選択信号PALUSEL (3:1)が有効となる。

【0104】

ここで実行時間測定器63の動作について図9のタイミングチャートを用いて説明する。

【0105】

$a-4$ の期間でクリティカルデータ生成器49によってALLF+0の演算を指定する。 $a-3$ から $a-1$ までの3サイクルの期間ALLF+1のクリティカルな演算を指定する。ALU20が期待値ALL0を出力した際にゲート67の出力AZ1がハイとなり、同様にALU21の出力がALL0になるとゲート80の出力AZ2が、ALU22の出力がALL0になるとゲート93の出力AZ3がハイとなる。AZ1は遅延バッファ69, 70, 71を経由して、AZ1D1、AZ1D2、AZ1D3となる。同様にAZ2は遅延バッファ82、83、84を経由して、AZ2D1、AZ2D2、AZ2D3となり、AZ3は遅延バッファ95、96、97を経由して、AZ3D1、AZ3D2、AZ3D3となる。

【0106】

RSFF68のNQ出力RES1はAZ2とAZ3から生成される信号をマスクしている。同様にRSFF81のNQ出力RES2はAZ1とAZ3から生成される信号をマスクしており、さらにRSFF94のNQ出力RES3はAZ1とAZ2から生成される信号をマスクしている。

【0107】

$a-3$ と $a-2$ の期間でALLF+1が実行され、ALUの能力に従ってAZ1、AZ2、AZ3および、RES1、RES2、RES3が生成される。AZ1、AZ2、AZ3の各信号線およびそれらの遅延信号とRES1、RES2、RES3の各マスク信号の生成順序にしたがって、ALU1SET1~4とALU2SET1~4とALU3SET1~4のうちでマスク信号より先にSET信号が変化した信号線がHIに変化する。本実施例においてはALU22が一番高速で、AZ3が一番早くHIに変化しており、それに従って、ALU3SET1

、ALU3SET2、ALU3SET3がHIに変化している。次にはAZ1がHIに変化し、それに従って、ALU1SET3がHIに変化している。以上の信号値から、PALUSEL(1)=0、PALUSEL(2)=0、PALUSEL(3)=1の値が出力されている。

【0108】

本実施形態の場合においてはPALUSEL(1)=0、PALUSEL(2)=0、PALUSEL(3)=1を反映して、セクタ28では通常はALU22の出力25を選択し、不一致信号34がハイの場合だけセクタ26の出力27を選択する。イネーブル信号64はFF37の出力39を出力するし、イネーブル信号65はFF37の出力39の反転信号を出力し、イネーブル信号66は常時ハイを出力する。

【0109】

なお、a以降の動作は実施形態1と同様となる。

【0110】

各ALUにおけるテストデータ実行時間測定の結果、最速のALUが決まると以下のように扱う

(1) ALU20が最高速の場合

PALUSEL(1)=1、PALUSEL(2)=0、PALUSEL(3)=0となり、セクタ28では通常はALU20の出力23を選択し、不一致信号34がハイの場合だけセクタ26の出力27を選択する。イネーブル信号64は常時ハイを出力し、イネーブル信号65はFF37の出力39の反転信号を出力し、イネーブル信号66はFF37の出力39を出力する。

【0111】

(2) ALU21が最高速の場合

PALUSEL(1)=0、PALUSEL(2)=1、PALUSEL(3)=0となり、セクタ28では通常はALU21の出力24を選択し、不一致信号34がハイの場合だけセクタ26の出力27を選択する。イネーブル信号64はFF37の出力39の反転信号を出力し、イネーブル信号65は常時ハイを出力し、イネーブル信号66はFF37の出力39を出力する。

【 0 1 1 2 】

なお、以上の説明では、各論理演算器のクリティカルパス実行時間とは、各論理演算器ユニット内部でのクリティカルパス演算処理時間として測定したが、クリティカルパス演算処理時間を各論理演算器ユニット内部での演算処理時間および各論理演算器ユニットからの出力結果を格納するレジスタ 3 1 に論理演算結果が格納されるまでの時間の和として測定しても良い。論理演算器の高速化進展により、論理演算器内部での遅延に加え、出力レジスタまでの配線遅延が無視できない場合も想定できるからである。この場合論理演算器内部での遅延と論理演算結果の出力レジスタまでの配線遅延の両者を加えてクリティカルパス演算処理時間の評価とする。

【 0 1 1 3 】

以上、情報処理装置において用意された複数の論理演算器のうち、クリティカルパスとなるテストデータを最も高速に実行できる論理演算器を検知して高速駆動の論理演算器として用いることにより、情報処理装置の処理効率の最大化を図ることができる。

【 0 1 1 4 】

【発明の効果】

以上のように、本発明に係る情報処理装置は、高速駆動の論理演算器と低速駆動の論理演算器の 2 種類の論理演算器を用い、高速駆動の論理演算器による高速処理により情報処理装置全体の処理速度向上を図ることができ、かつ、高速駆動の論理演算器では処理が間に合わないクリティカルパスのデータに対しては低速駆動の論理演算器により確実に処理を実行し、情報処理装置全体の正常処理を確保せしめることができる。

【 0 1 1 5 】

また、低速駆動の論理演算器を複数用意することにより、各低速駆動の論理演算器は高速なクロック周波数で駆動されているパイプラインの実行ステージを 1 つずつ順番に分担して担当し、たとえ自分が担当した実行ステージの演算がクリティカルパスであっても、それを十分処理できる低速駆動により演算を実行し、確実に正しい演算結果を得ることができ、パイプラインの駆動速度で高速駆動し

ている論理演算器の実行結果が正しいものか参照することができる。

【0116】

また、本発明に係る情報処理装置は、パイプラインの実行ステージの通常出力として前記高速駆動の論理演算器の出力をそのまま用いて通常のパイプライン処理を高速に実行し、かつ、演算処理結果が正しいものであるか否かを低速駆動の論理演算器の出力との比較により確認し、クリティカルパスなど正しい演算結果が得られていないものが検知されれば、低速駆動の論理演算器の出力で置き換えるという一種のエラー処理を行うことにより情報処理装置の正常動作を確保せしめることができる。

【0117】

また、本発明に係る情報処理装置は、情報処理装置が実際に処理するデータにおいて、クリティカルパスの発生頻度を調べることができ、クリティカルパスの発生頻度に応じてパイプラインを駆動するクロック周波数を変化させることができる。

【図面の簡単な説明】

【図1】 本発明の第1の実施形態の情報処理装置の構成を示すブロック図

【図2】 本発明の第1の実施形態の情報処理装置の動作を示すタイミング図

【図3】 本発明の第2の実施形態の情報処理装置の構成を示すブロック図

【図4】 本発明の第2の実施形態の情報処理装置により実行される各命令の実行時間の分布の一例を示す図

【図5】 (a) パイプライン駆動周波数を1ナノ秒増加させる調整を行った方が有利となる、オーバーフロー回数に対するオーバーフロー減少回数との関係を示す図、(b) パイプライン駆動周波数を1ナノ秒減少させる調整を行った方が有利となる、オーバーフロー回数に対するオーバーフロー増加回数との関係を示す図

【図6】 本発明の第3の実施形態の情報処理装置の構成を示すブロック図

【図7】 本発明の第3の実施形態の情報処理装置における実行時間測定器63の詳しい回路図

【図8】 本発明の第3の実施形態の情報処理装置の動作を示すタイミング図

【図 9】 図 8 のタイミング図の詳細を示す図

【図 1 0】 従来の情報処理装置の構成を示すブロック図

【図 1 1】 従来の情報処理装置の動作を示すタイミング図

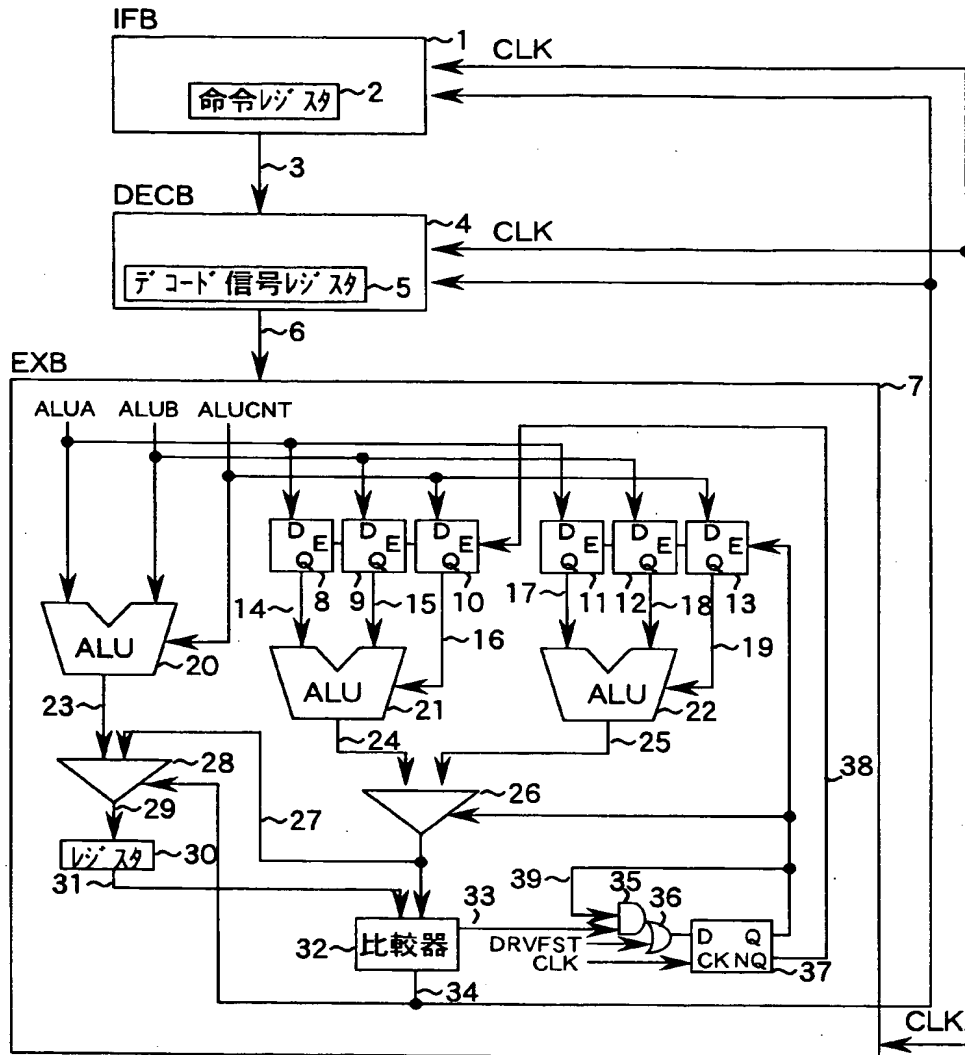
【符号の説明】

- 1 命令フェッチブロック
- 2 命令レジスタ
- 3 命令レジスタ出力
- 4 デコードブロック
- 5 デコード信号レジスタ
- 6 デコード信号レジスタ出力
- 7 実行ブロック
- 8, 9, 1 1, 1 2, 1 4, 1 5, 1 7, 1 8 ALU入力データラッチ
- 1 0, 1 3, 1 6, 1 9 ALU制御信号ラッチ
- 2 0, 2 1, 2 2 ALU
- 2 3, 2 4, 2 5 ALU出力
- 2 6, 2 8 セレクタ
- 2 7, 2 9 セレクタ出力
- 3 0 EXレジスタ
- 3 1 EXレジスタ出力
- 3 2 比較器
- 3 3 一致信号
- 3 4 不一致信号
- 3 5 ANDゲート
- 3 6 ORゲート
- 3 7 フリップフロップ
- 3 8 ALU入力イネーブル信号
- 3 9 ALU入力イネーブル信号
- 4 0 オーバーフロー検出器
- 4 1 検出期間設定レジスタ

- 4 2 最小検出回数設定レジスタ
- 4 3 最大検出回数設定レジスタ
- 4 4 クロックアップ要求信号
- 4 5 クロックダウン要求信号
- 4 6 クロック制御回路
- 4 7 クロック周波数初期値設定レジスタ
- 4 8 クロック周波数変化値設定レジスタ
- 4 9 クリティカルデータ生成器
- 5 0, 5 1, 5 2 クリティカル入力データ
- 5 3 クリティカル制御データ
- 5 4, 5 5, 5 6 セレクタ
- 5 7, 5 8, 5 9 セレクタ出力
- 6 0 A L U 入力データラッチ出力
- 6 1 A L U 入力データラッチ出力
- 6 2 A L U 制御信号ラッチ出力
- 6 3 実行時間測定器
- 6 4 A L U 入力イネーブル信号
- 6 5 A L U 入力イネーブル信号
- 6 6 A L U 入力イネーブル信号
- 6 7, 8 0, 9 3, 1 0 9, 1 1 3, 1 1 8 NORゲート
- 6 8, 7 6 ~ 7 9, 8 1, 8 9 ~ 9 2, 9 4, 1 0 2 ~ 1 0 5 R S F F
- 6 9, 7 0, 7 1, 8 2 ~ 8 4, 9 5 ~ 9 7 バッファ
- 7 2, 7 3, 7 4, 7 5, 8 5 ~ 8 8, 9 8 ~ 1 0 1, 1 0 6 ~ 1 0 8, 1 1 0 ~ 1 1 2, 1 1 4 ~ 1 1 7, 1 1 9 ANDゲート

【書類名】 図面

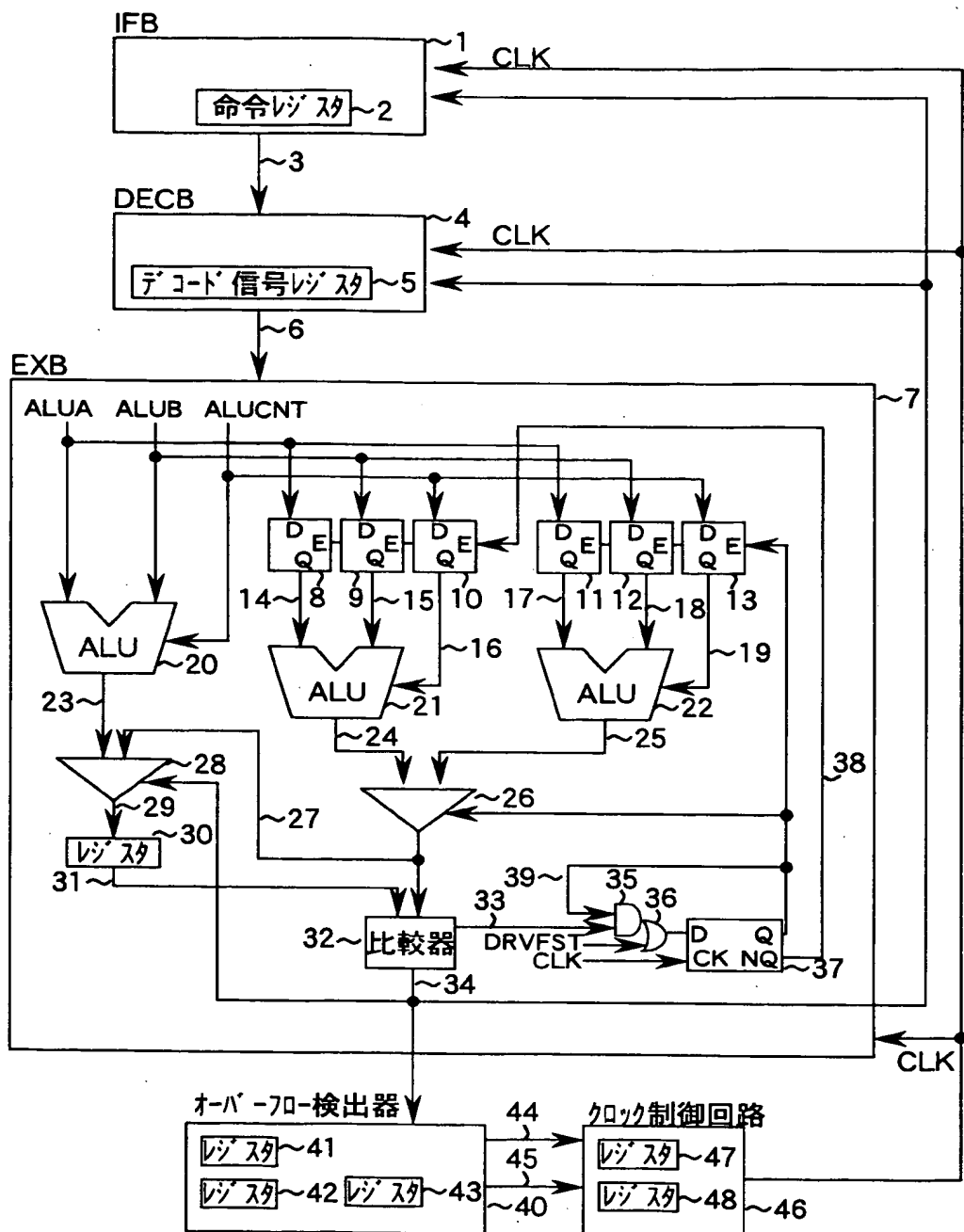
【図 1】



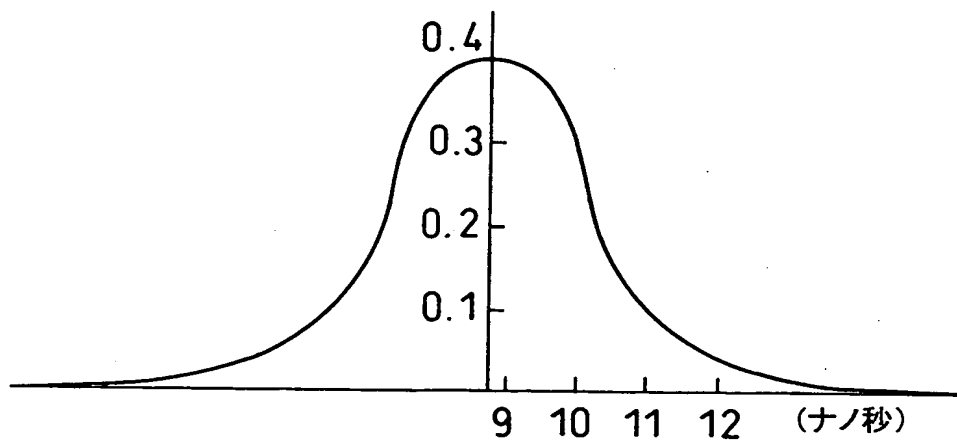
【図 2】

	a	b	c	d	e	f	g	h	i	j
CLK										
命令フィッチ	1	2	3	4	5	6	6	7	8	
命令レジスタ2		1	2	3	4	5	5	6	7	8
デコード		1	2	3	4	5	5	6	7	8
デコードレジスタ5			1	2	3	4	4	5	6	7
DECRV										
DRVFST										
EX			1	2	3	4	4	5	6	7
オペラ38										
レジスタ8, 9, 10			1		3			5		
ALU出力24			1		3		3'	5		
オペラ39										
レジスタ11, 12, 13				2		4			6	
ALU出力25				2		4			6	
セクタ出力27				1	2	3	3'	4	5	6
ALU出力23			1	2	3	4	4	5	6	7
EXレジスタ30				1	2	3	3'	4	5	6
EXRV										
一致信号33										
不一致信号34										

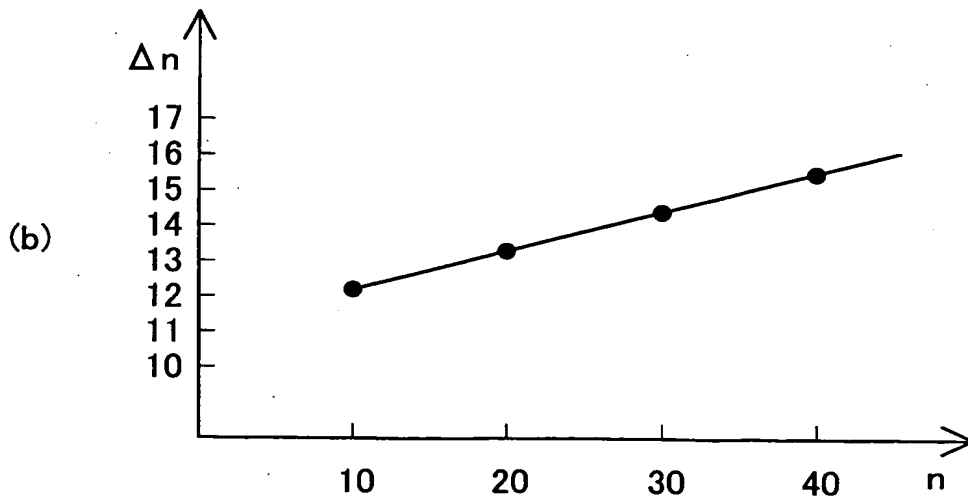
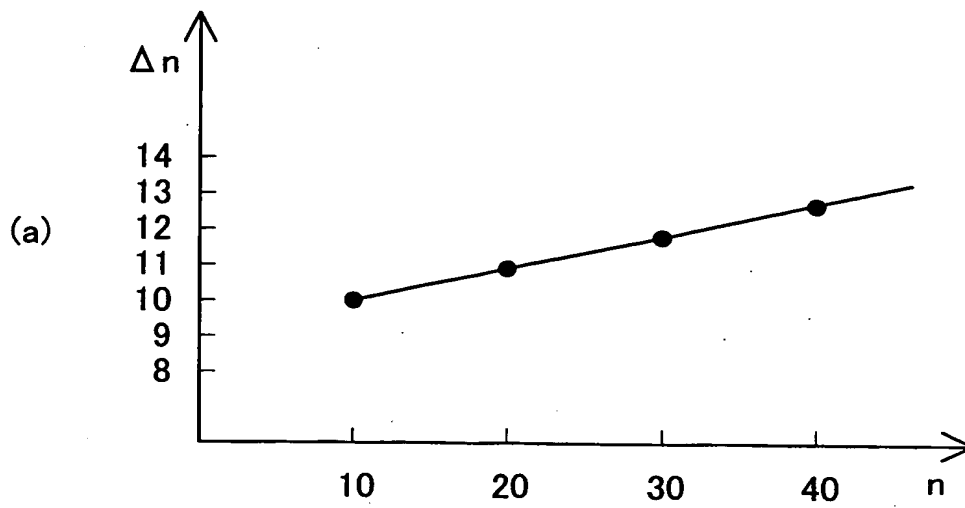
【図 3】



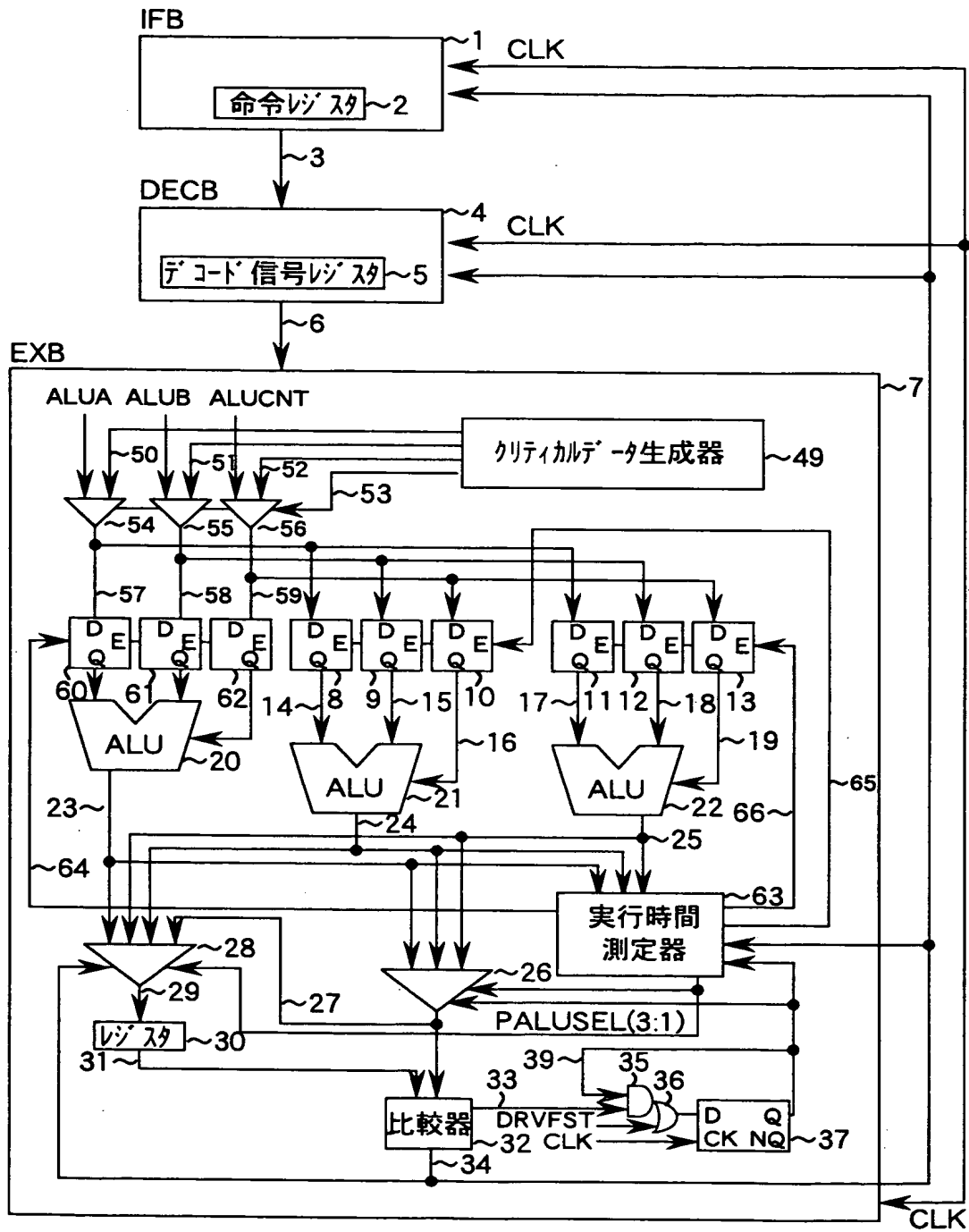
【図 4】



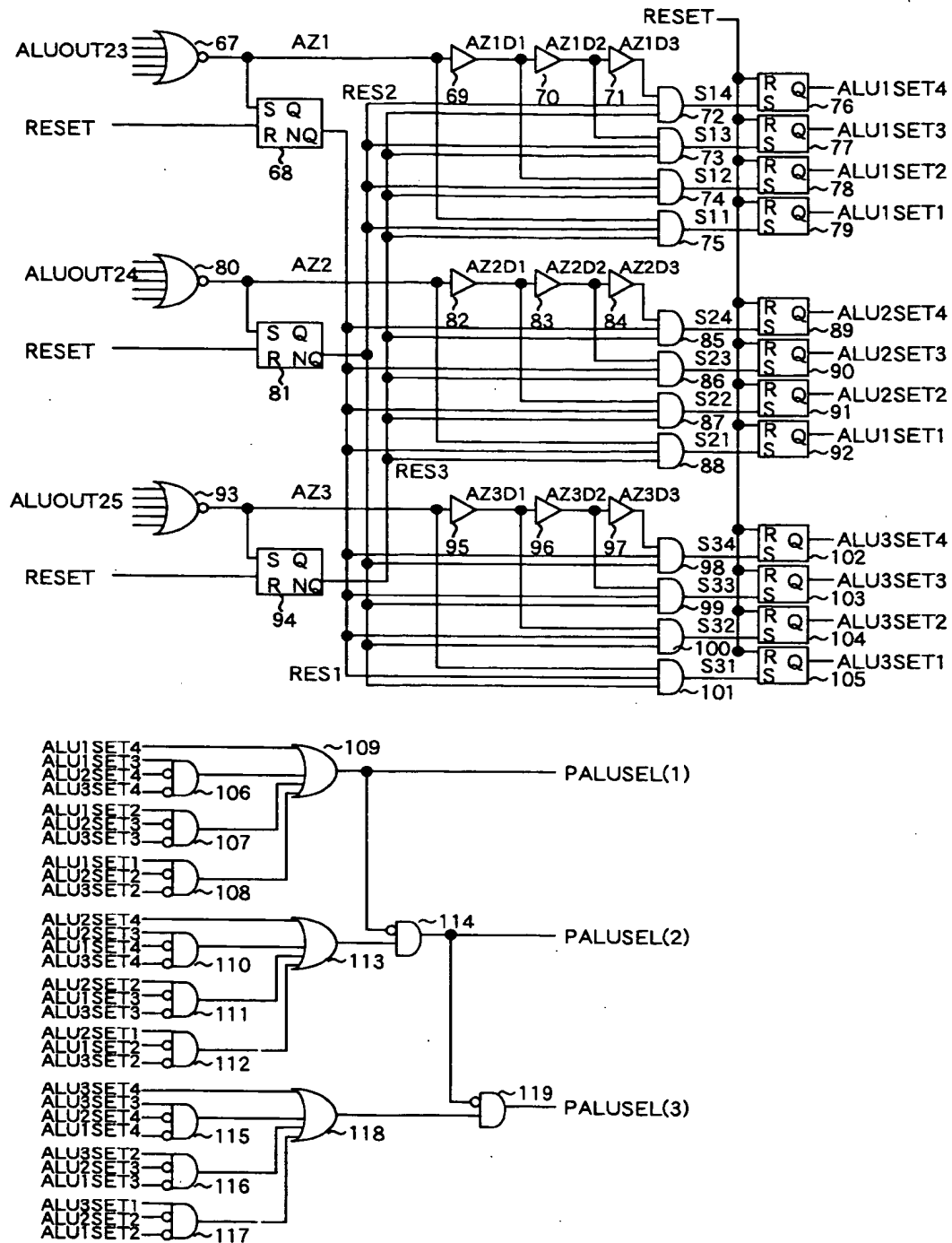
【図 5】



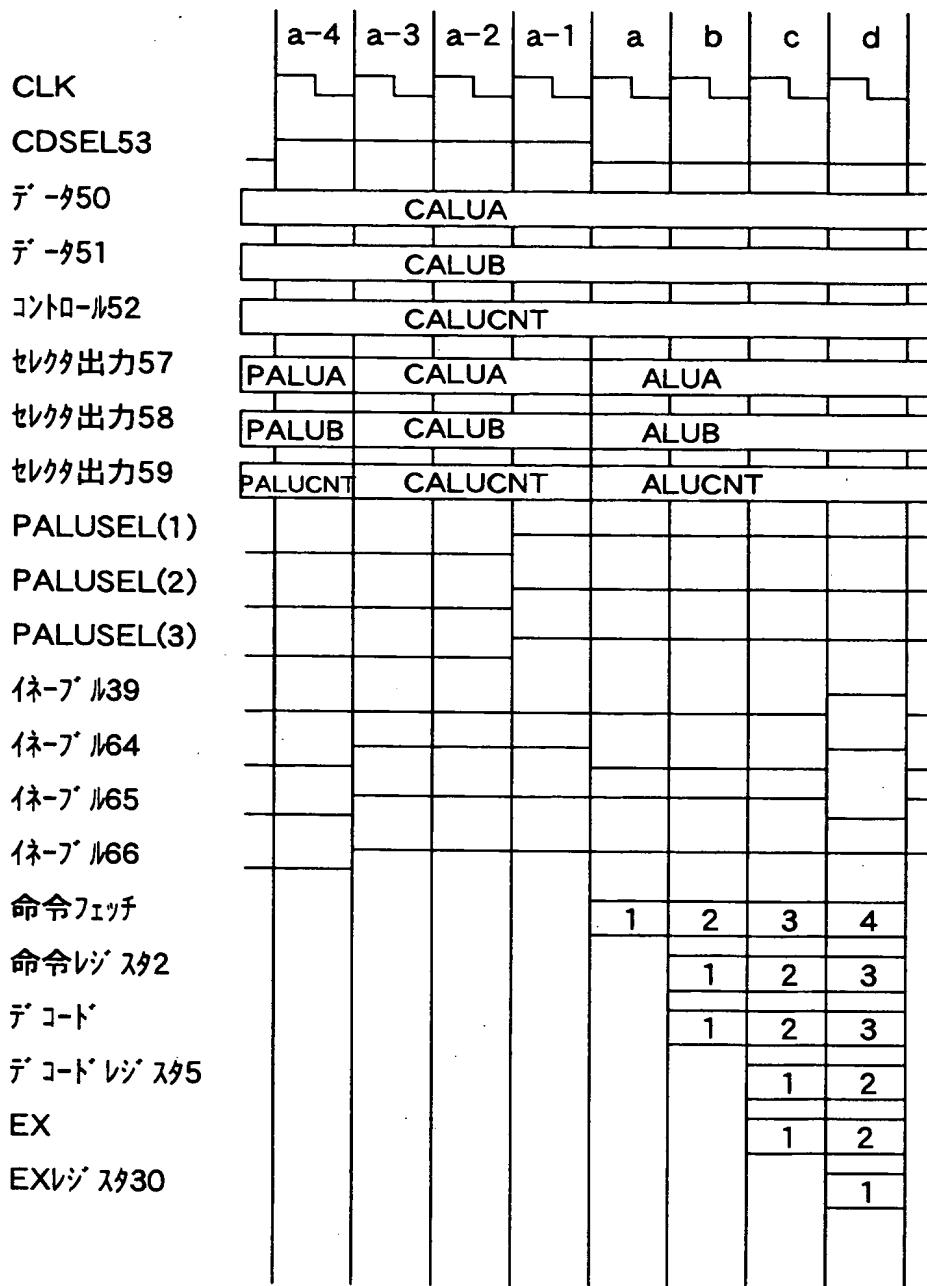
【図 6】



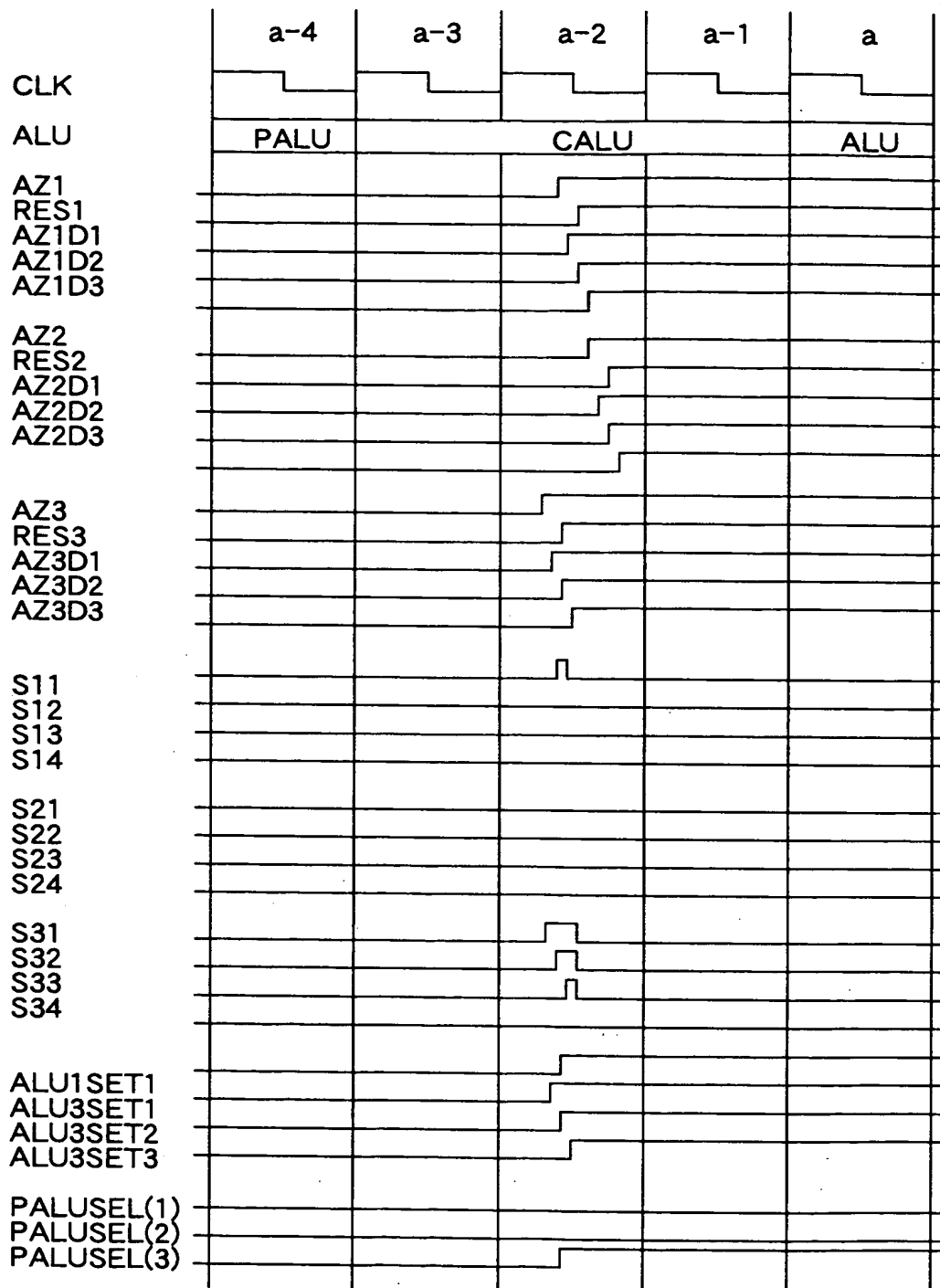
【図7】



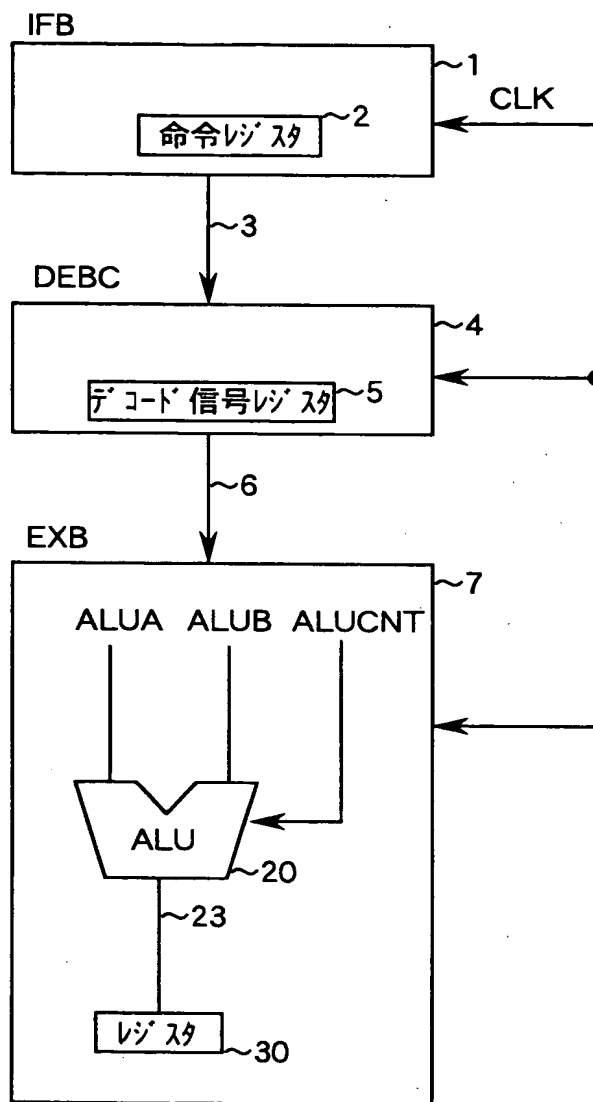
【図 8】



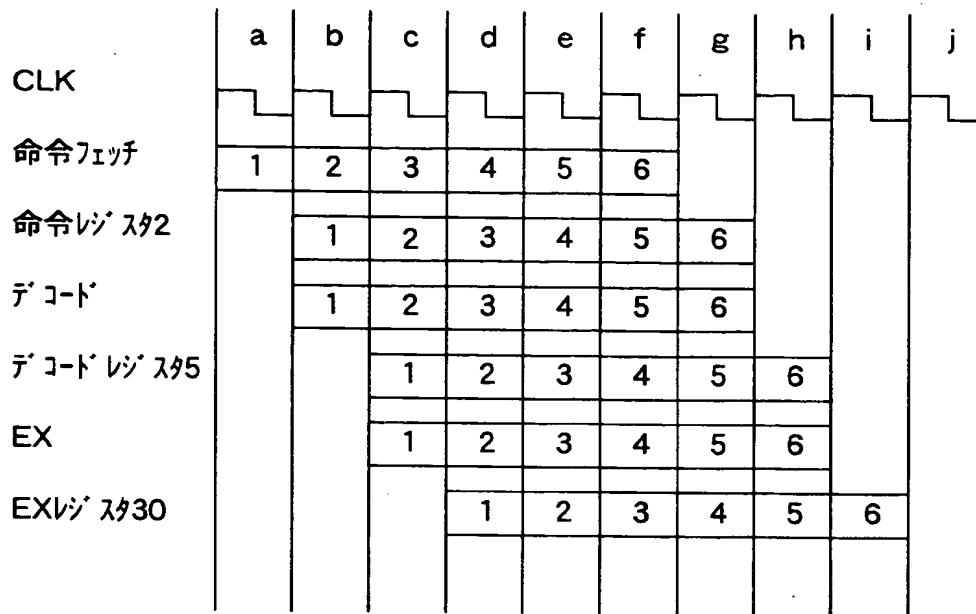
【図 9】



【図10】



【図 11】



【書類名】 要約書

【要約】

【課題】 クリティカルパス実行可能な周波数より高速なクロックでパイプラインを駆動できる情報処理装置を提供する。

【解決手段】 パイプラインを、クリティカルパスが処理できるクロック周波数より高速なクロック周波数で駆動する。パイプラインのクロック周波数で駆動する高速駆動のALU20に加え、クリティカルパスが処理できるクロック周波数で駆動する低速駆動のALU21とALU22を備え、各実行ステージの入力をALU21とALU22に交互に入力して2倍のマシンサイクルで演算を担当させる。高速駆動のALU20が正しく演算処理できた場合は、当該論理演算結果をパイプライン実行ステージの出力とし、高速駆動のALU20が正しく演算処理ができなかった場合は、該演算処理を担当した低速駆動のALU21またはALU22の演算結果をパイプライン実行ステージの出力とする。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [000005821]

1. 変更年月日 1990年 8月28日

[変更理由] 新規登録

住 所 大阪府門真市大字門真1006番地
氏 名 松下電器産業株式会社